参赛密码 _____

（由组委会填写）

“华为杯”第十三届全国研究生
数学建模竞赛

学　　校	华东师范大学
参赛队号	**K0209**
队员姓名	**1.**贾柯
	2.崔轩
	3.陈嘉骏

参赛密码 _____

（由组委会填写）

“华为杯”第十三届全国研究生数学建模竞赛

题 目　具有遗传性疾病和性状的遗传位点分析

摘　　要：

大量研究表明，人体的许多表型性状差异以及对药物和疾病的易感性等都可能与某些位点相关联，或和包含有多个位点的基因相关联。因此，定位与性状或疾病相关联的位点在染色体或基因中的位置，能帮助研究人员了解性状和一些疾病的遗传机理，也能使人们对致病位点加以干预，防止一些遗传病的发生。

对于问题一，根据位点中碱基对的特征，基于生物基因的加性效应，位点中的三种组合分别可编码为 0、1,、2，其中 1 代表杂合子基因，0 和 2 分别代表纯合子基因中的主要等位基因(major allele)与次要等位基因(minor allele)。

对于问题二，需通过一定方法计算出每个位点与疾病之间的关联程度，本文首先通过卡方检验进行建模，并且分别用 Benjamini & Hochberg（下简称 BH 校正）和 Bonferroni 校正（下简称 BONF 校正）P 值。阈值为经过校正后 P 值小于 0.05。满足阈值的致病相关位点为 **rs2273298（BH 校正后 p 值为 0.0006024;BONF 校正后 P 值为 0.0006024）。**除此之外，还采用置换检验模型和贝叶斯模型进行检验，检测出来最显著的致病位点与卡方检验相一致，因此最终得出与疾病相关的位点有一个，位点名称为：**rs2273298（置换检验模型校正后 p 值为 0.009445，贝叶斯因子的对数取值为 4.51238）。**

对于问题三，根据基因可以表示为位点的集合这一特征，本文采用 Set-based test 和 VEGAS 模型对一个基因内连锁不平衡的 SNP 位点进行建模，并且都采用置换算法进行模型求解，阈值为经过 BH 校正后 P 值小于 0.05，最

终两个模型得到了一致且较好的效果。与疾病相关联的基因有三个，基因所属序列为：gene_55、gene_102、gene_217（Set-based test 模型 BH 校正 P 值后分别为 0.149985,0.04,0.149985；VEGAS 模型 BH 校正后 P 值分别为 0.00165,0.0184,0.0009）。

对于问题四，多个性状往往表现为一个整体来进行衡量，本文分别采用 mv-plink 模型和 MultiPhen 模型对多个表型之间的关联进行建模，并且找出这些关联表型的致病位点。阈值选取为经过 BH 校正后 P 值小于 0.05。最终两个模型都得出与样本中十个性状有关联的位点有一个，位点名称为：rs12746773（mv-plink 模型 BH 校正后 P 值为 2.868447×10^{-20}；MultiPhen 模型 BH 校正后 P 值为 8.306198×10^{-21}）。

关键词：遗传统计学， 全基因组关联性分析(GWAS)，位点(SNPs) ，卡方检验

目 录

一、问题描述

人体的每条染色体携带一个 DNA 分子，人的遗传密码由人体中的 DNA 携带。DNA 是由分别带有 A,T,C,G 四种碱基的脱氧核苷酸链接组成的双螺旋长链分子。在这条双螺旋的长链中，共有约 30 亿个碱基对，而基因则是 DNA 长链中有遗传效应的一些片段。在组成 DNA 的数量浩瀚的碱基对（或对应的脱氧核苷酸）中，有一些特定位置的单个核苷酸经常发生变异引起 DNA 的多态性，我们称之为位点。染色体、基因和位点的结构关系见图 1-1。 在 DNA 长链中，位点个数约为碱基对个数的 1/1000。由于位点在 DNA 长链中出现频繁，多态性丰富，近年来成为人们研究 DNA 遗传信息的重要载体，被称为人类研究遗传学的第三类遗传标记。

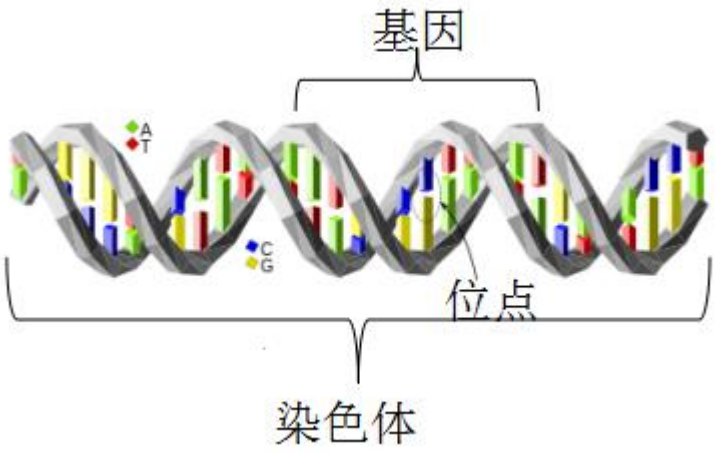

图 1-1 染色体、基因和位点的结构关系

大量研究表明，人体的许多表型性状差异以及对药物和疾病的易感性等都可能与某些位点相关联，或和包含有多个位点的基因相关联。因此，定位与性状或疾病相关联的位点在染色体或基因中的位置，能帮助研究人员了解性状和一些疾病的遗传机理，也能使人们对致病位点加以干预，防止一些遗传病的发生。近年来，研究人员大都采用全基因组的方法来确定致病位点或致病基因，具体做法是：招募大量志愿者（样本），包括具有某种遗传病的人和健康的人，通常用 1 表示病人，0 表示健康者。 对每个样本，采用碱基(A,T,C,G)的编码方式来获取每个位点的信息(因为染色体具有双螺旋结构，所以用两个碱基的组合表示一个位点的信息)。

本文提出的问题包括：

一、请用适当的方法，把 genotype.dat 中每个位点的碱基(A,T,C,G) 编码方式转化成数值编码方式，便于进行数据分析。

二、根据附录中 1000 个样本在某条有可能致病的染色体片段上的 9445 个位点的编码信息(见 genotype.dat)和样本患有遗传疾病 A 的信息（见 phenotype.txt 文件)。设计或采用一个方法，找出某种疾病最有可能的一个或几个致病位点，并给出相关的理论依据。

三、同上题中的样本患有遗传疾病 A 的信息（phenotype.txt 文件)。现有 300 个基因，每个基因所包含的位点名称见文件夹 gene_info 中的 300 个 dat 文件，每个 dat 文件列出了对应基因所包含的位点(位点信息见文件 genotype.dat)。由于可以把基因理解为若干个位点组成的集合，遗传疾病与基

因的关联性可以由基因中包含的位点的全集或其子集合表现出来请找出与疾病最有可能相关的一个或几个基因，并说明理由。

四、在问题二中，已知 9445 个位点，其编码信息见 genotype.dat 文件。在实际的研究中，科研人员往往把相关的性状或疾病看成一个整体，然后来探寻与它们相关的位点或基因。试根据 multi_phenos.txt 文件给出的 1000 个样本的 10 个相关联性状的信息及其 9445 个位点的编码信息(见 genotype.dat)，找出与 multi_phenos.txt 中 10 个性状有关联的位点。

二、合理假设与符号说明

2.1 合理假设

根据题意，可以进行如下假设：

1） 假设附件的样本均来自同一人种；

2） 性别对该疾病影响不大，即不考虑性别对疾病的影响；

3） 假设题目中所给数据都是有效数据；

2.2 符号说明

符号	定义
GWAS	全基因组关联分析
SNP	单核苷酸的多态性
H_0	原假设
H_1	备择假设
χ^2	卡方统计量
COV	协方差矩阵
T	置换检验次数

三、问题分析

3.1 问题一

问题一要求附件 genotype.dat 中的位点的碱基转化成数值编码。通过查阅相关资料可得，生物基因具有加性效应，并且同一个位点均表现为三种不同的碱基对，因此采用 0,1,2 的编码方式对位点碱基进行数值编码。

3.2 问题二

问题二要求从附录中 1000 个样本在某条有可能致病的染色体片段上的 9445 个位点的编码信息和样本患有遗传疾病 A 的信息中，找出相关的致病位点。针对这个问题，我们可能要采用检验独立性的模型去检验疾病与致病位点的关系。由于涉及到的假设检验次数很多，这就会形成多重假设检验问题。这就需要我们寻找 P 值校正的方法，从而找出可信度较高的致病位点。

3.3 问题三

问题三要求找出疾病 A 可能的致病基因。在进行单个 SNP 与疾病关联分析的时候，有许多的 SNP 与疾病的关系可能处于接近关联的程度，而这些 SNP 往往很容易被传统关联分析方法当作随机噪声而被舍弃掉。针对这样的情景，要根据一个基因内 SNP 间连锁不平衡的关系，把这些信号相对较低的 SNP 结合到基因的层级上，从而增强信号，检测出与疾病相关的基因。

3.4 问题四

问题四要求找出 10 个相关联性状的可能致病位点。在实际的研究过程中，往往把相关的性状或疾病看成一个整体。由于要关注性转间的关联，因此问题四的解决可能会用到回归类的模型以及典型显著分析模型，因为这两种模型都能较好地对变量间的关系进行建模。

四、模型特点介绍

4.1 问题二的建模

4.1.1 卡方检验模型

4.1.1.2 列联表的独立性检验模型

我们把按两个或多个特征分类的频数数据称为交叉分类数据，这些数据一般以表格的形式给出，称为列联表。一般，若总体中的个体可按两个属性 A 与 B 分类，A 有r个类，A_1，…，A_r，B 有c 个类B_1，…，B_c，从总体中抽取大小为 n 的样本，设其中有n_{ij}个个体既属于类A_i又属于B_j，n_{ij}称为频数，将$r\times c$个n_{ij}排列为一个r行c列的二维列联表，简称为$r\times c$列联表。若所考虑的属性多于两个，也可以按类似的方式作出列联表，称为多维列联表。

列联表分析的基本问题是，考虑各属性之间有无关联，即判别两属性是否独立。在$r\times c$列联表中，若以$p_{i.}$，$p_{.j}$和p_{ij}分别表示总体中的个体仅属于A_i，仅属于B_j和同时属于A_i和B_j的概率，可得到一个二维离散分布表，则“A、B 两属性独立”的假设可以表述为公式（4.1）：

$$H_0: p_{ij} = p_{i.}p_{.j}, i = 1,\ldots,r, j = 1,\ldots,c. \qquad (4.1)$$

这里诸p_{ij}共有rc个参数，在原假设H_0成立时，这rc个参数p_{ij}由$r+c$个参数

$p_{1.}$，…，$p_{r.}$和$p_{.1}$，…，$p_{.c}$决定，在这后$r+c$个参数中存在两个约束条件:$\sum_{i=1}^{r} p_{i.} = 1$, $\sum_{j=1}^{c} p_{.j}$所以，此时p_{ij}实际上由$r+c-2$个独立参数所确定的。据此，检验统计量为公式（4.2）

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c} \frac{\left(n_{ij} - n\hat{p}_{ij}\right)}{n\hat{p}_{ij}} \qquad (4.2)$$

在原假设H_0成立时上式近似服从自由度为$rc-(r+c-2)-1 = (r-1)(c-1)$的$\chi^2$分布，其中诸$\hat{p}_{ij}$是在$H_0$成立下得到的$p_{ij}$的最大似然估计，其表达式(4.3)

为

$$\hat{p}_{ij} = \hat{p}_{i.}\hat{p}_{.j} = \frac{n_{i.}}{n} \cdot \frac{n_{.j}}{n} \tag{4.3}$$

对给定的显著性水平α (0< α <1)，检验的拒绝域为公式（4.4）：

$$W = \left\{ \chi^2 \geq \chi^2_{1-\alpha}\left((r-1)(c-1)\right) \right\} \tag{4.4}$$

在针对疾病相关位点检测的时候，我们这样使用卡方检验：假设其中一个SNP位点基因型的分布情况为GG,GT,TT。它们在患病样本和非患病样本的分布如表4-1：

表4-1 患病和非患病样本分布表

	GG	GT	TT	总数
患病	r_0	r_1	r_2	R
非患病	s_0	s_1	s_2	S
总数	n_0	n_1	n_2	N

在进行卡方检验的时候，需要把表转化为表4-2：

表4-2 卡方检验转化表

	G	T	总数
患病	$2r_0 + r_1$	$r_1 + 2r_2$	2R
非患病	$2s_0 + s_1$	$s_1 + 2s_2$	2S
总数	$2n_0 + n_1$	$n_1 + 2n_2$	2N

4.1.1.2 Benjamini & Hochberg校正

多重假设检验的统计显著性问题已经引起了许多统计学者的注意。1995年，Benjamini和Hochberg在研究多重假设检验时首次从期望的角度提出了错误发现率的概念，并在多重检验中对它的控制方法做了研究，给出了计算方法。然而，由于当时没有学者研究大规模数据，因此并没有受到重视，甚至还受到广大学者的质疑。若干年后，随着微阵列数据研究的不断发展，大规模数据的频繁出现使得FDR有了实际的应用，错误发现率的理论和应用研究也在逐渐走向成熟。

FDR（False Discovery Rate）的定义如公式（4.5）：

$$FDR = \left\{ E\left(\frac{V}{V+S}\right) = E\left(\frac{V}{R}\right) \right. \tag{4.5}$$

上式中的V和S分别表示m个假设检验中错误拒绝和正确拒绝检验的个数，R表示m个假设检验中总的拒绝原假设的个数，$E(.)$表示数学期望[1]。

为了能够直观的得到接受或拒绝原假设的置信程度，我们通常采用P值来研究。在多重假设检验的研究中，采用P值进行假设检验已经成为国际上比较流行的方法。

1. P值的定义

设检验统计量为X，其样本观测值为x，对于一族拒绝域$\{\Gamma\}$，统计量X在

$X=x$ 处的 P 值可以定义为公式（4.6）

$$p(x)=\min_{\{\Gamma:x\in\Gamma\}}\{\Pr(X\in\Gamma\mid H_0)\} \tag{4.6}$$

在实际的假设检验中，由上述定义可以得到统计量在 $X=x$ 处的 P 值为 p，如果 $p\le 0.05$，说明检验结果是显著的，如果 $p\le 0.01$，则说明检验结果是非常显著的。

2. P 值的计算方法

当 H_0 为真时，统计量 X 的值 x 可由样本数据计算出，根据检验统计量 X 的实际分布，可以求出 P 值 $p(x)$。具体计算方法如下：

1) 左侧检验的 P 是统计量 X 小于样本统计量 x 的概率，即公式（4.7）

$$p(x)=P_r\{X<x\mid H_0\} \tag{4.7}$$

2) 右侧检验的 P 值是统计量 X 大于样本统计量 x 的概率，即公式（4.8）

$$p(x)=P_r\{X>x\mid H_0\} \tag{4.8}$$

3) 当统计量的分布具有对称的性质时（例如正态分布，t 分布等），双侧检验的 P 值是统计量 X 落在样本统计值 x 为断点的尾部取余内的概率的 2 倍，即当 x 位于分布曲线的右侧时，有公式（4.9）

$$p(x)=P_r\{|X|>|x|\mid H_0\}=2P_r\{X>x\mid H_0\} \tag{4.9}$$

当 x 位于分布曲线的左侧时，有公式（4.10）

$$p(x)=P_r\{|X|>|x|\mid H_0\}=2P_r\{X<x\mid H_0\} \tag{4.10}$$

4.1.1.3 Bonferroni 校正

两两比较中最常见的错误，是在每次比较中一直使以用总的检验标准（如α = 0.05），这样做的结果导致总的 I 类错误增大。如果α = 0.05 为水准分别对 m 个实际上成立的零假设（即实际情况是H_0为真的时候）进行检验，不犯 I 类错误的概率为 $(1-\alpha)_m$，至少出现一次错误的概率（ I 类错误的累积概率）为 $1-(1-\alpha)_m$，显然大于 0.05，例如 $m=3$ 时，$1-(1-\alpha)_m=0.14$ ，这么大的 I 类错误让人难以接受，且随着 m 增大, $1-(1-\alpha)_m$将更大。于是， Bonferroni 提出通过控制α，降低 I 类错误的累计概率：若每次检验水准为α ，共进行 m 次比较，当H_0为真时，犯 I 类错误的累积概率 α'不超过 mα，即有 Bonferroni 不等式（ Bonferroni inequality）$\alpha'\le m$α成立。所以令各次比较的检验水准为$\alpha'=0.05/m$，并规定 $P\le\alpha'$时拒绝H_0，基于这样的做法，就可以把 I 类错误的

累积概率控制在 0.05。这种对检验水准进行修正的方法叫做 Bonferroni 调整（ Bonferroni adjustment）法，简称 Bonferroni 法。Bonferroni 法公式（4.11）如下:

$$\alpha' = \frac{2\alpha}{k(k-1)} = \frac{\alpha}{m} \tag{4.11}$$

4.1.2 基于贝叶斯的 GWAS 模型

贝叶斯因子在基因位点关联性分析方面在此之前已经有大量的研究[2,3,4,5]，而且无论在可解释性方面还是在结果的准确性方面对比 P 值的方法均具有较好的优势。贝叶斯模型的主要过程是计算贝叶斯因子（BF）的值，即数据中位点与疾病相关（M_1）的边缘似然概率和位点与疾病不相关（M_0）的边缘似然概率之比，公式（4.12）表示如下：

$$BF = \frac{P(Data \mid M_1)}{P(Data \mid M_0)} \tag{4.12}$$

其中每个边缘概率定义为公式（4.13）：

$$P(Data \mid M_l) = \int \left(\sum_{i=1}^{N} \prod_{k=0}^{2} P(\Phi \mid G_{ij} = k, \theta) p_{ijk} \right) P(\theta \mid M_l) d\theta \tag{4.13}$$

其中 i 代表 N 个不同的样本，j 代表样本所属的第 j 个位点，并且有 $G_i \in \{0,1,2\}$，代表第 i 个样本第 j 个位点的位点类型。边缘概率的计算可采用 Laplace 近似的方法得到。这一步我们同时可以计算得到相应观测数据似然概率的极值点和对应的先验θ：

$$\hat{\theta} = \arg\max_{\theta} \left(\prod_{i=1}^{N} \sum_{k=0}^{2} P(\Phi \mid G_{ij} = k, \theta) p_{ijk} \right) P(\theta \mid M_1) \tag{4.14}$$

因此相应的打分和信息矩阵表示为公式（4.15）和（4.16）：

$$U^*(\theta) = E_{Y_M \mid Y_O, \theta}[U(\theta)] + \frac{d \log P(\theta \mid M_l)}{d\theta} \tag{4.15}$$

$$I^*(\theta) = E_{Y_M \mid Y_O, \theta}[I(\theta)] - V_{Y_M \mid Y_O, \theta}[U(\theta)] - \frac{d^2 \log P(\theta \mid M_l)}{d\theta^2} \tag{4.16}$$

对于后验概率近似的极致 $\hat{\theta}_{M_1}$ 可以通过 Newton-Raphson 迭代来获得，也可以通过 EM 算法不断迭代获取。

对于只有两种性状（得病和没得病）的数据模型来说，如果我们想要计算M_1的边缘似然概率，首先要选取适当的先验分布，关于先验的选取 Stephens and Balding[5]详细的分析和讨论，这里我们使用 $P(\theta|M_1)=P(\mu)P(\gamma)$ 作为先验分布，其中 $\mu\sim N(0,1),\gamma\sim N(0,s^2),s=0.2$。因此可以得到公式（4.17）和（4.18）：

$$\frac{d\log P(\theta|M_1)}{d\theta}=(-\mu\ -\gamma s^{-2})^T, \tag{4.17}$$

$$\frac{d^2\log P(\theta|M_1)}{d\theta^2}=\begin{pmatrix}-1 & 0\\ 0 & -s^{-2}\end{pmatrix}. \tag{4.18}$$

M_0的边缘似然概率可以通过相同的方法得到。因此可以得到对于疾病相关和疾病无关的边缘似然概率近似贝叶斯因子值为公式（4.19）和（4.20）：

$$\log P(Data|M_1)\approx(\sum_{i=1}^{N}\sum_{k=0}^{2}P(\Phi|G_{ij}=k,\hat{\theta}_{M_1})p_{ijk})+\log P(\hat{\theta}_{M_1}|M_1)+\log(2\pi)-\frac{1}{2}\log|I^*(\hat{\theta}_{M_1})| \tag{4.19}$$

$$\log P(Data|M_0)\approx(\sum_{i=1}^{N}\sum_{k=0}^{2}P(\Phi|G_{ij}=k,\hat{\theta}_{M_0})p_{ijk})+\log P(\hat{\theta}_{M0}|M_0)+\log(2\pi)-\frac{1}{2}\log|I^*(\hat{\theta}_{M_0})| \tag{4.20}$$

Stephens and Balding 中指出采用正态分布作为先验太小对 GWAS 来说而不能真实反映实际的影响值，因此他提出使用混合正态分布来增加先验值，其中一个方法是对参数γ使用 t 先验分布，t 分布概率密度如公式（4.21）：

$$f(\gamma;m,s,d)=\frac{\Gamma(\frac{(d+1)}{2})}{\Gamma(\frac{d}{2})d^{1/2}\pi^{1/2}s}\left[1+\frac{(\gamma-m)^2}{ds^2}\right]^{-\frac{d+1}{2}} \tag{4.21}$$

其中 m 为样本均值，s^2为样本方差，d 为自由度，如果保持$\mu\sim N(0,1)$那么可得公式（4.22）和（4.23）：

$$\frac{d\log P(\theta|M_l)}{d\theta}=\left(-\mu-\frac{(d+1)(\gamma-m)}{ds^2-(\gamma-m)^2}\right)^T, \tag{4.22}$$

$$\frac{d^2 \log P(\theta \mid M_l)}{d\theta^2} = \begin{pmatrix} -1 & 0 \\ 0 & -\dfrac{(d+1)(ds^2+(\gamma-m))}{ds^2-(\gamma-m)^2} \end{pmatrix}^T \quad (4.23)$$

4.1.3 置换检验模型

模拟运算法的基本原理是：根据所研究的问题构造一个检验统计量，并利用样本，按排列组合的原理，导出检验统计量的理论抽样分布；若难以导出确切的理论分布，则采用抽样模拟的方法做估计其近似分布。然后求出从该分布中获得样本及更极端样本的概率（P 值），并界定此概率值作出推论。若检验统计量的抽样分布是基于样本的所有可能的排列(或组合）条件下的分布，则称之为“Exact Permutation Test（EPT）”，可译为“确切排列（组合）检验”，其思路类似于秩和检验。对实际问题来说，往往得不到检验统计量的确切抽样分布，可通过基于样本的大量重复的随机排列(或组合）估计其近似的抽样分布，则称之为“Randomized Permutation test（RPT）”，可译为“随机排列(或组合）检验”。

在全基因组关联分析中，采用标签置换和基因抛除是采用模拟运算法的两个基本做法。一般采用自适应的模拟运算法来估计理论抽样分布。在自适应算法中，对于没有太大重要性的位点会比看起来有一些重要性的位点更快地被抛弃。也就是说，如果在十次对于某一个位点的模拟运算过程中有 9 次统计值都大于观测到的抽样统计，那么再继续对这个位点模拟运算也没有什么必要了，因为这个位点已经很不可能会显现出比别的位点更加显著的结果。这个步骤可以大大加快模拟运算的过程，因为在刚开始的阶段大量没有意义的位点都会被抛弃掉，这样可以使得模拟运算到有意义的位点上的概率大大增加。因此很自然地，那些对疾病有意义的位点统计得到的 P 值会比那些对于疾病没有太大意义的位点得到的 P 值更加准确，但是这并不对我们的结果造成影响。

自适应标签置换检验模型中一共有 6 个参数，参数含义如表 4-3：

表 4-3 参数含义

参数表示	默认值	参数涵义
γ	5	每一个位点的最小置换次数
θ	1000000	每一个位点的最大置换次数
α	0	α级别阈值
β	0.0001	经验 P 值的置信区间
μ	1	位点调整区间
τ	0.001	位点调整区间减慢速度

上面参数表示：对于每一个位点，至少要经过γ次置换才能进行位点的调整，并且最多不能超过θ次置换。μ表示经过$\mu*\gamma$次之后进行位点调整，并且下一次调整间隔次数增加为$\mu*\gamma+0.001R$次，R 是当前进行过的置换次数。在每一次的置换过程中，对每一个期望 P 值的置信区间为公式（4.24）：

$$CI=\left(100*\left(\frac{1-\beta}{2T}\right)\right)\% \tag{4.24}$$

其中 T 为位点个数。

相应流程图如图所示：

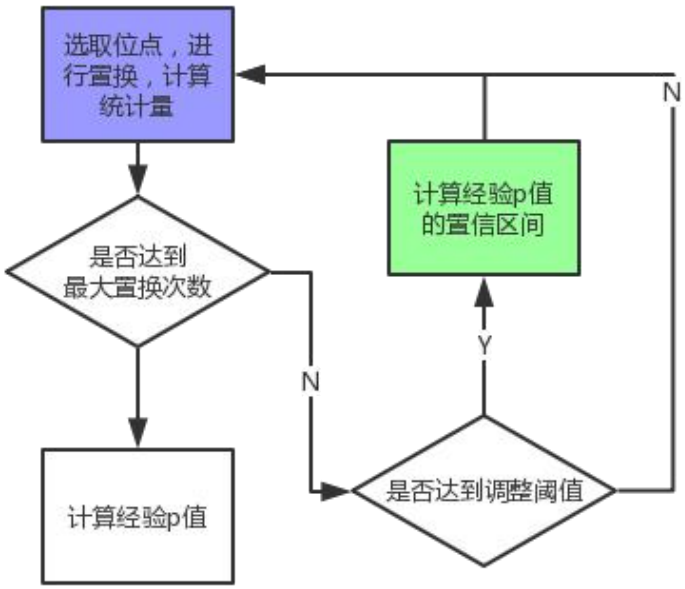

图 4-1 置换检验模型流程图

4.2 问题三的建模

4.2.1 基于集合的基因检验模型（Set-based test）

两个 SNP 之间的连锁不平衡是用来度量两个 SNP 位点之间的相关程度。其中 D 和r平方量化两个 SNP 位点之间连锁不平衡程度的指标。在这里，基因座 L1 位点为 A 的概率为p_A,基因座 L1 位点为 a 的概率为q_a;基因座 L2 位点为 B 的概率为p_B，基因座 L2 位点为 b 的概率为q_b。因此我们有下表 4-4：

表 4-4

	L2 B	L2 b
L1 A	$P_{AB} = p_A p_B + D$	$P_{Ab} = p_A q_b - D$
L1 a	$P_{aB} = q_a P_B - D$	$P_{AB} = q_a q_b + D$

由上表我们可以得到公式（4.25）和（4.26）：

$$D = P_{AB}P_{ab} - P_{Ab}P_{aB} \tag{4.25}$$

$$r^2 = \frac{D^2}{p_A q_a p_B q_b} \tag{4.26}$$

该模型的过程如下：

步骤一：对于每一个基因，我们检测每个 SNP 与其他 SNP 的是否连锁不平衡，检验得到的 r 若超过阈值 R 的时候，则认为两个 SNP 之间是连锁不平衡的。

步骤二：对于每一个 SNP，我们进行单个 SNP 和疾病之间的关联分析，在这里我们选用卡方检验来进行单个 SNP 与疾病之间的关联分析。

步骤三：对于每个基因，我们选取 N 个独立的 SNP（来源于步骤一的连锁不平衡检验），P 值小于步骤一中的阈值 P。首先选取 P 值最小的那个 SNP，去除与选定 SNP 连锁不平衡的其他的 SNP，根据统计量的显著程度进行排列。

步骤四：对于这些 SNP 的子集，每个基因的统计量由基因中的 SNP 的统计量的均值表示。

步骤五：对数据集进行大量的置换，保证 SNP 之间的连锁不平衡状态不变。

步骤六：对于每个被置换的数据集，重复步骤二和步骤四。

步骤七：基因的经验 P 值计算，由置换后基因的统计量超过未经置换的基因的统计量的次数除以所有的置换次数。

4.2.2 全面基于基因关联分析模型(VEGAS)

基于基因的关联分析模型在进行关联分析的时候，通常考虑的是一个基因内的所有 SNP，而不是像传统的关联分析那样只考虑一个 SNP。根据基因组得而组织结构来看，基于基因的全基因组关联分析会比传统基于单个 SNP 的关联分析效果更加好。例如，当一个基因包含多个致病的 SNP，而这几个在相同基因里的 SNP，在进行单个 SNP 的关联分析的时候，可能只是处在接近显著的程度，而 SNP 和可能会被当成随机的噪音被舍弃掉。当把这些同一个基因中的致病 SNP 结合到一起进行相关的统计检验和连锁不平衡方面的校正，基于基因的方法可能就会检测出这些接近显著的 SNP 相关的基因。

下面是 VEGAS 模型的过程：

步骤一：对于一个基因中的 n 个 SNP，产生一个 n 维向量 d，该向量服从正态分布，该正态分布的均值为 0，方差矩阵为$\sum$ 。其中Σ为$n \times n$矩阵，该矩阵的元素为成对连锁不平衡中的 r 值

步骤二：对矩阵$\sum$ 进行 Cholesky 矩阵分解，我们得到$n\times n$下三角矩阵 C，其中$\sum = CC^{\Gamma}$ 。

步骤三：然后用向量 d 与下三角矩阵进行相乘，得到 n 维向量$Q_{original}$

步骤四：产生一个新的 n 维向量$Z=(z_1,\ldots,z_n)$, Z 服从正态分布$N_n(0,\sum\ )$, 然后把向量转化为自由量为 1 的相关卡方变量，$Q_{new}=(q_1,\ldots,q_n), q_i=z_i^{\ 2}$。

步骤五：统计量为 n 维向量 Q 中元素的和。

步骤六：对数据进行置换，重复步骤四和步骤五

步骤七：基因的经验 P 值计算，由置换后基因的统计量超过未经置换的基因的统计量的次数除以所有的置换次数。

4.3 问题四的建模

4.3.1 基于典型关联分析的多表型模型（MV-Plink）

典型关联分析是利用综合变量对之间的相关关系来反应两组指标之间的整体相关性的多元统计分析方法。它的基本原理是：为了把握两组指标之间的相关关系，分别在两组变量中提取有代表性的两个综合变量，利用两个综合变量之间的相关关系来反映两组指标之间的整体相关性。

典型关联分析在这里的应用主要是为了寻找表型 Y 和基因型 X 之间的相关关系最大化。这里，$\hat{p}=corr(Y,X)$成为典型相关系数估计值。为了得到$\hat{p}$，矩阵 Y 和矩阵 X 的协方差矩阵会被分块为如公式（4.27）：

$$COV\begin{bmatrix}X\\Y\end{bmatrix}=\begin{bmatrix}\sum_{YY} & \sum_{YX}\\ \sum_{XY} & \sum_{XX}\end{bmatrix} \qquad (4.27)$$

其中$\sum_{YY}$ 为矩阵 Y 的$K\times K$ 方差矩阵，$\sum_{XX}$ 为矩阵 X 的$K\times K$ 方差矩阵，$\sum_{YX}$ 和$\sum_{XY}$ 为矩阵 X 和矩阵 Y 的$K\times 1$或者$1\times K$ 协方差矩阵。

通过上述的矩阵分块，可以计算出矩阵$B=\sum_{YY}^{-1}\sum_{YX}\sum_{XX}^{-1}\sum_{XY}$ ，$\hat{p}$ 为矩阵 B 最大的特征值开平方根，$\hat{p}=\dfrac{\sum_{XY}a}{\left(a^{\Gamma}\sum_{YY}a\sum_{XX}\right)^{\frac{1}{2}}}$其中向量$a$为矩阵 B 最大的特征值锁对应的特征向量。此处基于典型关联分析多表型模型采用 Wiki lambda

$\lambda=1-\hat{p}^2$和 Wiki lambda 相关的 F 检验来获取 P 值。其中 F 统计量为公式

（4.28）
$$F = \frac{\hat{p}^2}{K\left(1-\hat{p}^2\right)\left(n-K-1\right)} \quad (4.28)$$

4.3.2 MultiPhen 模型

针对第四问，我们试图从单个表型信号接近显著的 SNP 中结合多个表型，然后找出信号显著的 SNP。在这里我们使用 MultiPhen 模型，从多个相关的表型中提炼出具有强相关联的 SNP。

在标准的多元 GWAS 方法分析中，通常使用线性回归对表型 Y 和基因型 X 进行相关的建模，其中我们令$Y_i = \{Y_{i1}, \ldots, Y_{ik}\}$,其中 i 为第 i 个人，k 表示第 k 个表型，令$X_i = \{X_{i1}, \ldots, X_{ig}\}$,其中$X_{ig} = \{0,1,2\}$，i 表示第 i 个人，g 表示第 g 个基因型。通过回归的方法来检测第 g 个基因型 SNP 和第 k 个表型之间的关系，我们有如下公式（4.29）：

$$Y_{ik} = \alpha_k + \beta_{gk} X_{ig} + \varepsilon_{igk} \quad (4.29)$$

其中ε_{igk}表示残差，而且假设服从正态分布。而在 MultiPhen 模型中所采用的方法是逆回归的方法，具体来说就是把基因型 X 作为因变量，表型 Y 作为自变量。基因型数据 X 是等位基因的数量，需要用到序数回归来进行相关的建模，这里使用的是比例逻辑回归（proportional odds logistic regression），这个模型的类别概率定义如公式(4.30)：

$$\mathrm{P}\left(X_{ig} \le m\right) = \frac{1}{1 + e^{-(\alpha_{gm} - \sum_{k=1}^{K} \beta_{gk} Y_{ik})}} \quad (4.30)$$

其中对于每一个 SNP，g=1,...,G 使用似然比率检验（likelihood ratio test）去检验原假设$\beta_{g1} = \cdots = \beta_{gk} = 0$。这个检验不用假设 Hardy-Weinberg 平衡。

五、问题求解

5.1 问题一求解

根据本题数据特征，每个位点均表示为两个不同碱基的组合形式，并且由于染色体的双螺旋结构，每种位点只有三种组合方式，如在位点 rs100015 位置，不同样本的编码都是 T 和 C 的组合，有三种不同编码方式 TT,TC 和 CC。因此基于生物基因的加性效应 (Additive genetic effects[6])，位点中的三种组合分别可编码为 0、1,、2，其中 1 代表杂合子基因，0 和 2 分别代表另外纯合子基因中的主要等位基因(major allele)与次要等位基因(minor allele)。主要等位基因和次要等位基因主要由频率决定，频率较低的为次要等位基因。比如，在题中给出的位点样本中，前十列位点对应的基因编码方式表 5-1：

表 5-1 问题一基因编码方式表

位点名称	次要/主要等位基因	次要等位基因（纯合子）	杂合子	主要等位基因（纯合子）
rs3094315	C/T	CC => 2	CT/TC => 1	TT => 0
rs3131972	T/C	TT => 2	CT/TC => 1	CC => 0
rs3131969	T/C	TT => 2	CT/TC => 1	CC => 0
rs1048488	C/T	CC => 2	CT/TC => 1	TT => 0
rs12562034	A/G	AA => 2	AG/GA => 2	GG => 0
rs12124819	G/A	GG => 2	GA/AG => 2	AA => 0
rs4040617	G/A	GG => 2	GA/AG => 2	AA => 0
rs2980300	T/C	TT => 2	CT/TC => 1	CC => 0
rs4970383	T/G	TT => 2	TG/GT => 1	GG => 0
rs4475691	T/C	TT => 2	CT/TC => 1	CC => 0

5.2 问题二求解

分析题目可得本题主要过程是建立计算样本中位点与样本整体之间的关联程度，即独立性程度，因此本题采用卡方检验模型。在进行卡方检验之后，我们得到了 9445 次检验的 P 值，然后作出相关的 Q-Q 图。如图 5-1，可以发现部分 P 值的实际值与观测值存在较大的偏差。在 GWAS 研究中，则认为这个 SNP 位点的观测值的偏离是由这个 SNP 突变所产生的。这些检验说明可能存在致病性的 SNP 位点。下文中我们选用适合的阈值，对 SNP 致病性位点作进一步的筛选。

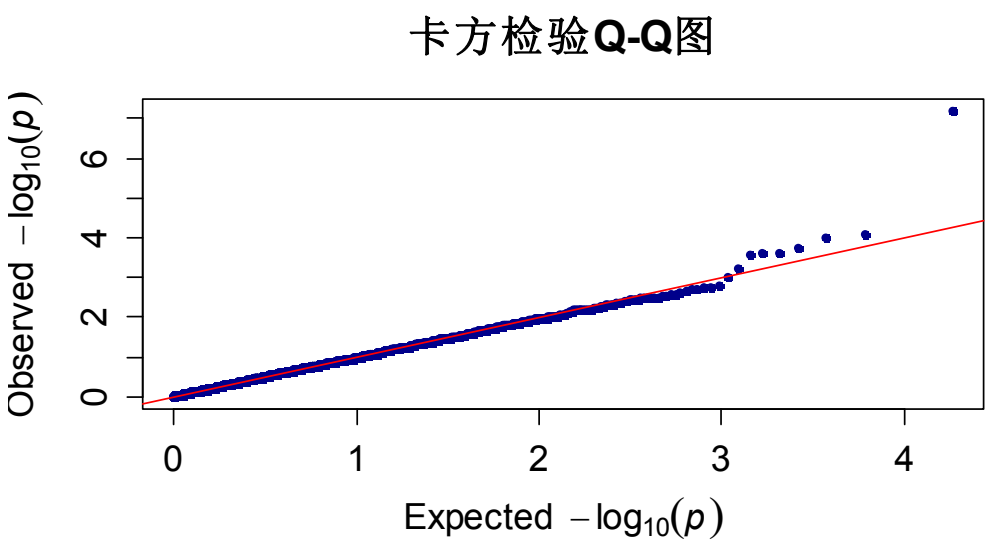

图 5-1 问题二 卡方检验 Q-Q 图

由于问题二中涉及到的位点有 9445 个，也就是说要进行 9445 次的卡方检验，这就涉及到了多重检验的问题，因此需要对 P 值的阈值进行调整或者对 P 值本身进行校正，要不然会产生假阳性。我们在对 P 值进行阈值调整的时候选用了 $\frac{0.05}{T}$ 新的阈值，在对 P 值本身进行校正方面，还选用了 BH 校正以及 BONF 校正。从图 5-2 中，我们可以看出 BONF 校正相对于 BH 校正来说，更加的保守。除了在图 5-2 的结果以外，也有其他论文报道过 BH 进行校正的时候，能够尽可能在减少假阳性的同时，避免假阴性增加太多，因此下文所有的 P 值校正均采用 BH 校正。分别使用置换检验模型与贝叶斯模型进行结果的统计分析与检验，由表 5-2，表 5-4，表 5-5 的结果对比表明这两种模型得到的实验结果很好地印证了卡方检验的结果，因此最终选取与某种疾病最相关的位点为：rs2273298。通过卡方检验模型得到的曼哈顿图如图 5-2 所示：

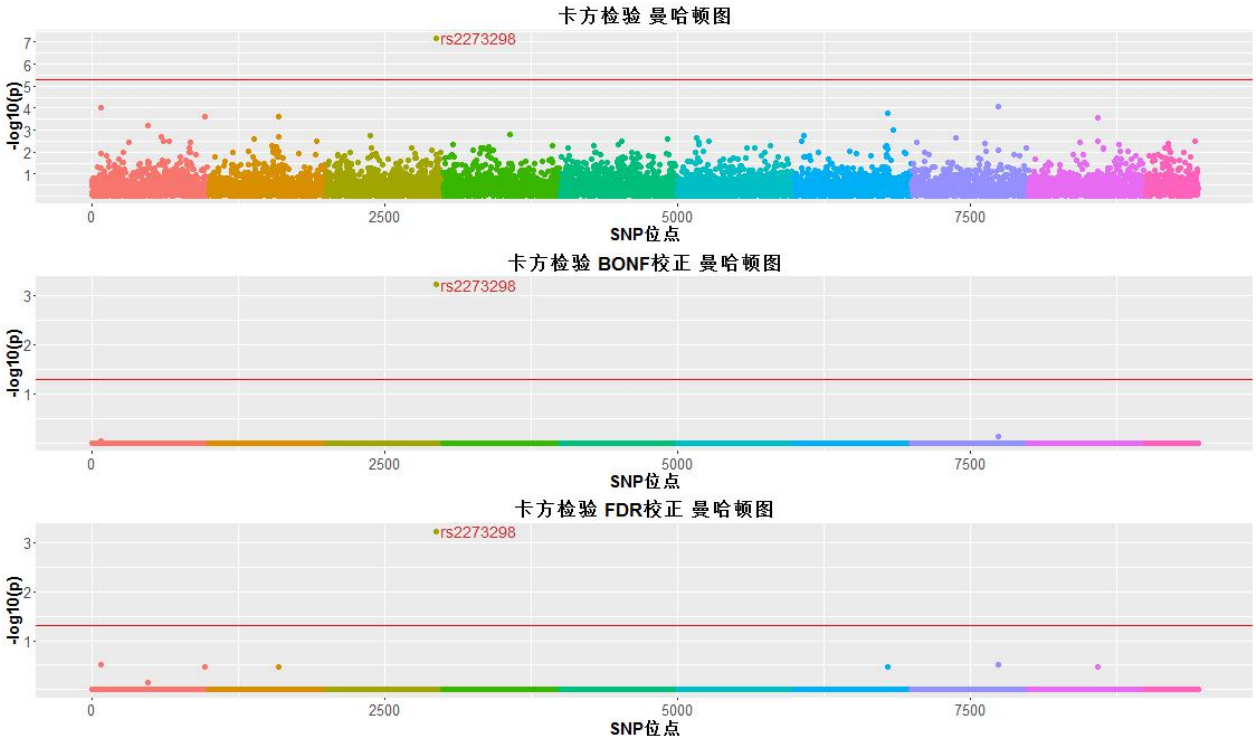

图 5-2 问题二 卡方检验的曼哈顿图

卡方检验模型得到的前 10 个与某种疾病相关的位点关联程度如表 5-2 所示：

表 5-2 问题二 位点关联表

P 值 / 位点	卡方检验模型		
	调整前	BH 调整	BONF 调整
rs2273298	6.378e-8	0.0006024	0.0006024
rs932372	8.028e-5	0.3013	0.7582
rs12036216	9.571e-5	0.3013	0.904
rs2807345	0.0001738	0.3485	1
rs4391636	0.0002414	0.3485	1
rs7522344	0.00025	0.3485	1
rs9426306	0.0002583	0.3485	1
rs12133956	0.0006022	0.7109	1
rs11580218	0.000964	0.987	1
rs590368	0.001576	0.987	1

置换检验模型采用的参数如表 5-3 所示：

表 5-3 问题二 参数列表

参数表示	参数值
γ	10
ϑ	1000000
α	0.0001
β	0.01
μ	5
τ	0.001

相关参数含义参考模型介绍。最终置换检验模型得到的图 5-2。

置换检验模型得到的前 10 个与某种疾病相关的位点关联程度如表 5-4 所示：

表 5-4 问题二 置换检验的位点关联表

期望 P 值 / 位点	置换检验模型	
	调整前	调整后
rs2273298	1e-6	0.009445
rs932372	7.6e-5	0.311685
rs12036216	9.9e-5	0.311685
rs4391636	0.0001978	0.4298824
rs9426306	0.0002982	0.4298824
rs7522344	0.0003134	0.4298824
rs2807345	0.0003186	0.4298824
rs11580218	0.0007781	0.8693598
rs12133956	0.0008284	0.8693598
rs707472	0.001186	1

贝叶斯模型得到的前 10 个与某种疾病相关的位点关联程度如表 5-5 所示：

表 5-5 问题二 贝叶斯模型的位点关联表

log10_bf / 位点	贝叶斯模型
rs2273298	4.51238
rs932372	2.13662
rs4391636	2.12278
rs12036216	2.04464
rs7522344	2.03471
rs9426306	2.01492
rs2807345	1.9341
rs12133965	1.65489
rs11580218	1.57159
rs707472	1.4864

5.3 问题三求解

根据问题三种数据的特征，基因与疾病的关联性可由基因相关的位点的关联性表示，因此本题采用基于集合的基因检验模型与 VEGAS 模型来对样本进行建模求解，最终得到与疾病关联最有可能的基因为：gene_102、gene_55、gene_217。

采用基于集合的基因检验模型与 VEGAS 模型对基因样本进行关联性分析结果如图 5-3 所示：

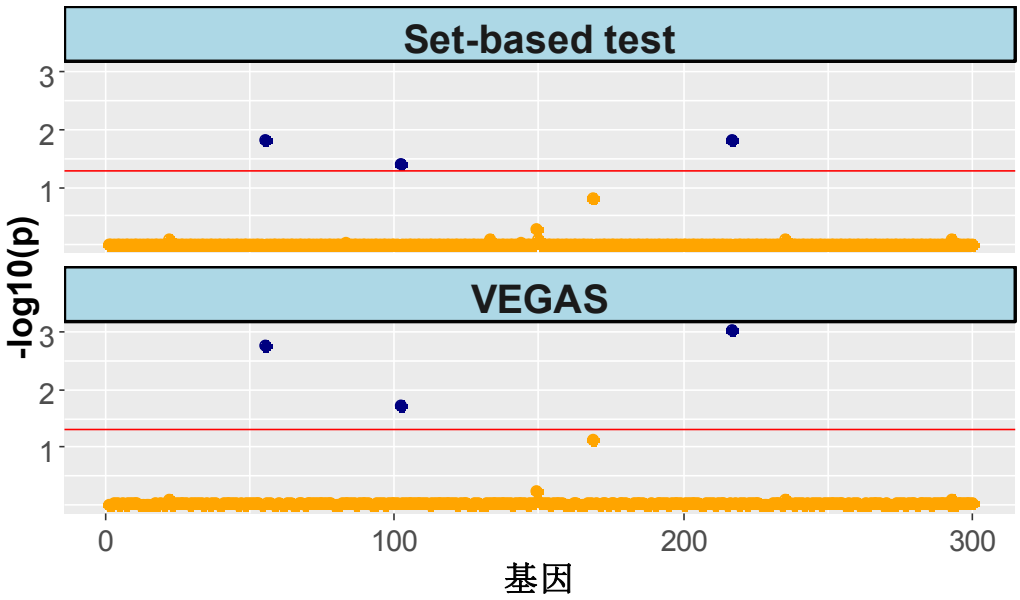

图 5-3 问题三 关联性分析结果图

其中两种模型前 10 个（两个模型前 10 排序相同）与疾病相关的基因 BH 校正 p 值如表 5-6 所示：

表 5-6 问题三 两个模型的 BH 校正 P 值对比表

校正 P 值 / 基因	VEGAS 模型	Set-based 检验模型
gene_217	0.0009	0.0149985
gene_55	0.00165	0.0149985
gene_102	0.0184	0.04
gene_169	0.0759	0.15
gene_149	0.5574	0.53394
gene_22	0.7992857	0.7885714
gene_293	0.7992875	0.7885714
gene_235	0.8156250	0.822
gene_150	0.8773333	0.822
gene_133	0.9234	0.822

5.4 问题四求解

针对问题中多个性状的特征，采用 mv-plink 模型和 MultiPhen 模型进行问题求解。阈值选取为 BH 校正过后，P 值小于 0.05。两个模型均在样本中保持较好且较一致的效果，最终得到与样本中 10 个性状有关联的位点为：rs12746773。

两个模型对应的 BH 校正曼哈顿图如图 5-4 所示：

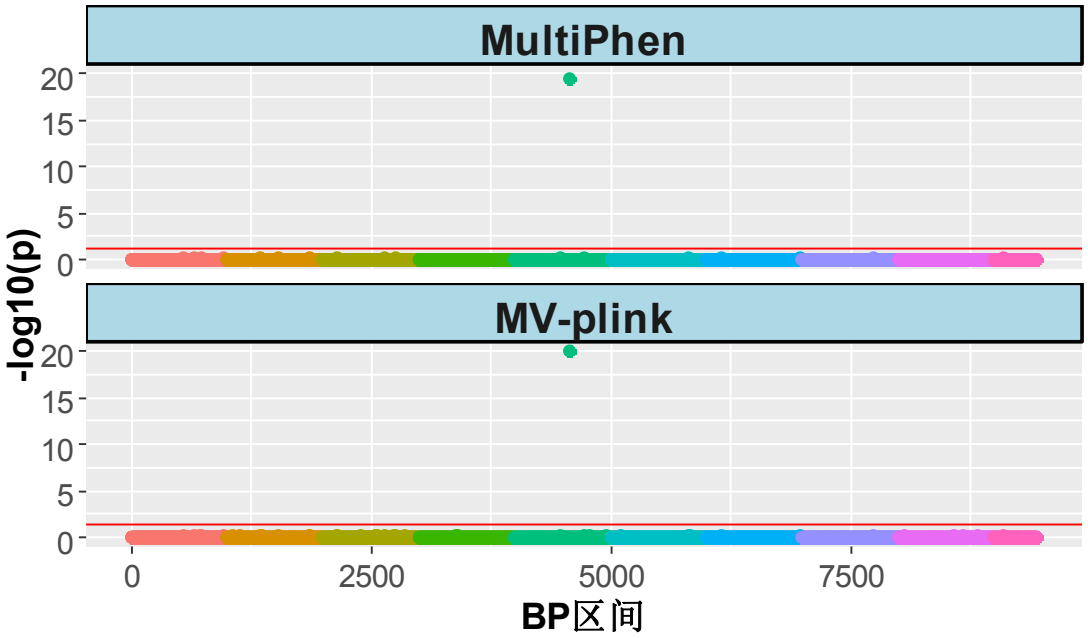

图 5-4 问题四 BH 校正曼哈顿

其中两种模型前 10 个（两个模型前 10 排序相同）与疾病相关的基因 BH 校正 P 值如表 5-7 所示：

表 5-7 问题四 两个模型的 BH 校正 P 值对比表

校正 P 值 / 位点	mv-plink	校正 P 值 / 位点	MultiPhen
rs12746773	2.868447e-20	rs12746773	8.306198e-21
rs2483275	5.408352e-1	rs10916755	5.772251e-1
rs2185639	5.408352e-1	rs10157835	5.772251e-1
rs10915423	5.408352e-1	rs4908877	5.772251e-1
rs4908877	5.408352e-1	rs12144375	5.772251e-1
rs12144375	5.408352e-1	rs10916755	5.772251e-1
rs4908803	5.408352e-1	rs11249248	5.772251e-1
rs629524	5.408352e-1	rs10799128	5.772251e-1
rs7549599	5.408352e-1	rs11577262	5.772251e-1
rs10916755	5.408352e-1	rs11811497	6.199957e-1

六、模型评价

本文针对问题中给出的数据的结构特点，建立不同的模型进行数据分类，并综合运用各种方法得到较好效果。同时为了满足不同数据的需求，不断对模型进行改进提升，主要工作如下：

1. 针对本题并没有独立的数据集进行验证，问题二、问题三、问题四，我们都给出了至少两种的模型，并且通过这些不同模型的结果进行相互的验证。
2. 针对单个 SNP 与疾病进行关联分析的时候，接近显著的 SNP 往往会被忽略，这里我们利用同一基因中 SNP 可能的连锁不平衡关系，找出与疾病相关的基因。并且对 VEGAS 模型中的基因列表以及基因与 SNP 的关系进行精简，使得模型更加适合题目的应用场景。

参考文献

[1]. Benjamin Y,Hochberg Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing[J]. Statist Society Series, 1995,57(2):289-300.

[2]. Scott, L., Mohlke, K., Bonnycastle, L., Willer, C., Li, Y., Duren, W., Erdos, M., Stringham, H., Chines, P., Jackson, A., Prokunina-Olsson, L., Ding, C., Swift, A., Narisu, N., Hu, T., Pruim, R., Xiao, R., Li, X., Conneely, K., Riebow, N., Sprau, A., Tong, M., White, P., Hetrick, K., Barnhart, M., Bark, C., Goldstein, J., Watkins, L., Xiang, F., Saramies, J., Buchanan, T., Watanabe, R., Valle, T., Kinnunen, L., Abecasis, G., Pugh, E., Doheny, K., Bergman, R., Tuomilehto, J., Collins, F., and Boehnke, M. A genome-wide association study of type 2 diabetes in finns detects multiple susceptibility variants. Science 316(5829), 1341－5 (2007).

[3]. Guan, Y. and Stephens, M. Practical issues in imputation-based association mapping. PLoS Genet 4(12), e1000279 (2008).

[4]. Wakefield, J. Bayes factors for genome-wide association studies: comparison with p-values. Genetic Epidemiology 33, 79－86 (2009).

[5]. Stephens, M. and Balding, D. Bayesian statistical methods for genetic association studies. Nat Rev Genet 10(10), 681－690 (2009).

[6]. https://en.wikipedia.org/wiki/Additive_genetic_effects.

[7]. Ferreira M A R, Purcell S M. A multivariate test of association[J]. Bioinformatics, 2009, 25(1): 132-133.

[8]. Galesloot T E, Van Steen K, Kiemeney L A L M, et al. A comparison of multivariate genome-wide association methods[J]. PloS one, 2014, 9(4): e95923.

[9]. O'Reilly P F, Hoggart C J, Pomyen Y, et al. MultiPhen: joint model of multiple phenotypes can increase discovery in GWAS[J]. PloS one, 2012, 7(5): e34861.

[10]. Purcell S, Neale B, Todd-Brown K, et al. PLINK: a tool set for whole-genome association and population-based linkage analyses[J]. The American Journal of Human Genetics, 2007, 81(3): 559-575.

[11]. Liu J Z, Mcrae A F, Nyholt D R, et al. A versatile gene-based test for genome-wide association studies[J]. The American Journal of Human Genetics, 2010, 87(1): 139-145.

附件

R 绘图代码：

第二题曼哈顿图：

```
ata=read.csv("D:\\math\\chi_square_2_questiones\\result_chi_square.csv")
library(dplyr)
data_new=dplyr::mutate(data,index=seq(1,dim(data)[1],1),CHR=rep(1,dim(data)[1]),bp=factor(c(rep(1:9,each=1000),rep(10,445))))
library(ggplot2)
p1=ggplot(data_new,aes(x=data_new$index,y=-log10(data_new$P),colour=bp))+
  geom_point(size=2)+
  guides(fill=FALSE)+
  scale_y_continuous(breaks=c(seq(1,10)))+
  xlab("SNP 位点")+
  ylab("-log10(p)")+

  theme(axis.title.x=element_text(colour                    ="black",face="bold",size=15))+
  theme(axis.title.y=element_text(colour                    ="black",face="bold",size=15))+
  theme(axis.text.x=element_text(size=rel(1.5)))+
  theme(axis.text.y=element_text(size=rel(1.5)))+
  ggtitle("卡方检验 曼哈顿图")+
  theme(plot.title=element_text(size  =  rel(1.5),lineheight  =0.9,colour = "black",face="bold"))+
  geom_hline(yintercept = -log10(0.05/9945),colour="red")+

annotate("text",x=3300,y=7.195179,label="rs2273298",size=5,colour="firebrick3")+
  theme(legend.position='none')

p2=ggplot(data_new,aes(x=data_new$index,y=-log10(data_new$BONF),colour=bp))+
  geom_point(size=2)+
  guides(fill=FALSE)+
  scale_y_continuous(breaks=c(seq(1,4)))+
  xlab("SNP 位点")+
  ylab("-log10(p)")+

  theme(axis.title.x=element_text(colour                    ="black",face="bold",size=15))+
  theme(axis.title.y=element_text(colour                    =
```

```
"black",face="bold",size=15))+
      theme(axis.text.x=element_text(size=rel(1.5)))+
      theme(axis.text.y=element_text(size=rel(1.5)))+
      ggtitle("卡方检验 BONF 校正 曼哈顿图")+
      theme(plot.title=element_text(size  =  rel(1.5),lineheight  =
0.9,colour = "black",face="bold"))+
      geom_hline(yintercept = -log10(0.05),colour="red")+

annotate("text",x=3300,y=3.22,label="rs2273298",size=5,colour="fireb
rick3")+
      theme(legend.position='none')

    p3=ggplot(data_new,aes(x=data_new$index,y=-log10(data_new$FDR_BH)
,colour=bp))+
      geom_point(size=2)+
      guides(fill=FALSE)+
      scale_y_continuous(breaks=c(seq(1,4)))+
      xlab("SNP 位点")+
      ylab("-log10(p)")+

      theme(axis.title.x=element_text(colour                          =
"black",face="bold",size=15))+
      theme(axis.title.y=element_text(colour                          =
"black",face="bold",size=15))+
      theme(axis.text.x=element_text(size=rel(1.5)))+
      theme(axis.text.y=element_text(size=rel(1.5)))+
      ggtitle("卡方检验 FDR 校正 曼哈顿图")+
      theme(plot.title=element_text(size  =  rel(1.5),lineheight  =
0.9,colour = "black",face="bold"))+
      geom_hline(yintercept = -log10(0.05),colour="red")+

annotate("text",x=3300,y=3.22,label="rs2273298",size=5,colour="fireb
rick3")+
      theme(legend.position='none')



    grid.newpage()
    pushViewport(viewport(layout = grid.layout(3,1)))
    vplayout  =  function(x,  y)  viewport(layout.pos.row  =  x,
layout.pos.col = y)
    print(p1, vp = vplayout(1, 1))
    print(p2, vp = vplayout(2, 1))
    print(p3, vp = vplayout(3, 1))
```

第三题曼哈顿图：

```
library(ggplot2)
library(dplyr)
data_plink=read.csv("D:\\math\\question3\\order\\result_all.csv",
sep=" ")
data_vegas=read.csv("D:\\math\\question3\\order\\result_vegas.cs
v",sep=" ")
data_plink_new=dplyr::mutate(data_plink,gene=seq(1,300))
data_vegas_new=dplyr::mutate(data_vegas,gene=seq(1,300))
data_vegas_new$Pvalue=p.adjust(data_vegas_new$Pvalue,method    =
"fdr")
data_plink_new$EMP1=p.adjust(data_plink_new$EMP1,method = "fdr")
colnames(data_vegas_new)=c("GENE_NAME","pvalue","gene")
colnames(data_plink_new)=c("GENE_NAME","pvalue","gene")
result=rbind(data_plink_new,data_vegas_new)
method=c(rep("Set-based test",300),rep("VEGAS",300))
significant=ifelse(result$pvalue<0.05,"yes","no")
result_final=cbind(result,method,significant)
ggplot(result_final,aes(x=result_final$gene,y=-log10(result_fina
l$pvalue),colour=result_final$significant))+
  geom_point(size=2)+
  facet_wrap(~method,nrow = 2)+
  scale_colour_manual(values=c("orange","navy"))+
  scale_y_continuous(breaks=c(seq(1,4)))+
  xlab("基因")+
  ylab("-log10(p)")+

  theme(axis.title.x=element_text(colour                          =
"black",face="bold",size=15))+
  theme(axis.title.y=element_text(colour                          =
"black",face="bold",size=15))+
  theme(axis.text.x=element_text(size=rel(1.5)))+
  theme(axis.text.y=element_text(size=rel(1.5)))+
  #ggtitle("曼哈顿图")+
  theme(plot.title=element_text(size   =   rel(1.5),lineheight   =
0.9,colour = "black",face="bold"))+
  geom_hline(yintercept = -log10(0.05),colour="red")+

#annotate("text",x=3300,y=3.22,label="rs2273298",size=5,colour="fire
brick3")+
  theme(legend.position='none')+
  theme(strip.text=element_text(face="bold",size=rel(1.5)),

strip.background=element_rect(fill="lightblue",colour="black",size=1)
```

```
)
```

第四题曼哈顿图：

```
data_multiphen=read.csv("D:\\math\\question_4\\multiphen.csv")
data_mvplink=read.csv("D:\\math\\question_4\\mvplink.csv")
data_mvplink$P=p.adjust(data_mvplink$P,method="fdr")
data_multiphen$JointModel=p.adjust(data_multiphen$JointModel,method="fdr")
colnames(data_multiphen)=c("SNP","P")
result=rbind(data_mvplink,data_multiphen)
method=c(rep("MV-plink",9445),rep("MultiPhen",9445))
BP=rep(factor(c(rep(1:9,each=1000),rep(10,445))),2)
index=rep(seq(1:9445),2)
result_final=cbind(result,method,index,BP)

ggplot(result_final,aes(x=result_final$index,y=-log10(result_final$P),colour=BP))+
  geom_point(size=2)+
  facet_wrap(~method,nrow=2)+
  #scale_y_continuous(breaks=c(seq(1,4)))+
  xlab("BP 区间")+
  ylab("-log10(p)")+

  theme(axis.title.x=element_text(colour="black",face="bold",size=15))+
  theme(axis.title.y=element_text(colour="black",face="bold",size=15))+
  theme(axis.text.x=element_text(size=rel(1.5)))+
  theme(axis.text.y=element_text(size=rel(1.5)))+
  #ggtitle("GWAS 卡方检验 BONF 校正 曼哈顿图")+
  theme(plot.title=element_text(size = rel(1.5),lineheight = 0.9,colour = "black",face="bold"))+
  geom_hline(yintercept = -log10(0.05),colour="red")+

#annotate("text",x=3300,y=3.22,label="rs2273298",size=5,colour="firebrick3")+
  theme(legend.position='none')+
  theme(strip.text=element_text(face="bold",size=rel(1.5)),

strip.background=element_rect(fill="lightblue",colour="black",size=1)
)
```

基因位点集合预处理：

```
import sys
prefix=sys.argv[1]
outfile = open(sys.argv[2],'w')
```

```
for i in range(1,301):
    inputfile = open(("%s%d.dat" % (prefix,i)),'r')
    outfile.write(("GENE_%03d" % (i)))
    for snp in inputfile:
        outfile.write(" "+snp[:-1])
    outfile.write(" END\n")
    inputfile.close()

outfile.close()
```

基因位点集合样本预处理

```
import sys

genedir = sys.argv[1]
geneorg = open(sys.argv[2],'r')
line = geneorg.readline()
splits = line[:-1].split(" ")
index=0
rsmap={}
for sp in splits:
    rsmap[sp]=index
    index+=1

pedfile = open(sys.argv[3],'r')
mapfile = open(sys.argv[4],'r')
genersmap=[]
for line in mapfile:
    genersmap.append(line)
outdir = sys.argv[5]
genetypes=[]
for line in pedfile:
    splits = line.split(" ")
    genetypes.append(splits)

GENE_NUM=300
for i in range(1,GENE_NUM+1):
    genefile = open(("%sgene_%d.dat") % (genedir,i),'r')
    genename = ("GENE_NO_%d" % (i))
    sinmapfile=open(outdir+genename+".map",'w')
    rsindex=[]
    for line in genefile:
        rsindex.append(rsmap[line[:-1]])
        sinmapfile.write(genersmap[rsmap[line[:-1]]])
    sinmapfile.close()
```

```
    outgeneped = open(outdir+genename+".ped",'w')
    for genetype in genetypes:
        outgeneped.write(" ".join(genetype[:5]))
        for ind in rsindex:
            cols = ind
            outgeneped.write(" "+genetype[cols+5])
        outgeneped.write("\n")
    outgeneped.close()
```

SNPtest 样本数据预处理

```
#!/usr/bin/python
import sys

genofile = open(sys.argv[1],'r')
phenofile = open(sys.argv[2],'r')

genfile = open(sys.argv[3],'w')
samplefile = open(sys.argv[4],'w')

type_number=0
line_number=0
phenotypes=[]
snps=[]
genotypes=[]
genotypemaps=[]
for line in phenofile:
    splits = line[:-1].split(" ")
    phenotype=[]
    for sp in splits:
        phenotype.append(int(sp))
    type_number = len(phenotype)
    phenotypes.append(phenotype)
print "Sample Size: %d." % (len(phenotypes))
for line in genofile:
    splits=line[:-1].split(" ")
    if line_number==0:
        snps=splits
    else:
        genotypes.append(splits)
    line_number+=1
print "Done %d lines data." % (line_number)
row_number = len(genotypes[0])
for i in range(0,row_number):
    snp=snps[i]
```

```
        typemap={}
        for j in range(0,line_number-1):
            sp = genotypes[j][i]
            if sp[0]!=sp[1]:
                typemap[2]=sp
            else:
               if 1 in typemap and typemap[1]!=sp:
                    typemap[3]=sp
               else:
                    typemap[1]=sp
            if len(typemap)==3:
                break
        genfile.write("SNP%04d     %s     10000     %s     %s"     %
(i,snp,typemap[1][0],typemap[3][0]))
        for j in range(0,line_number-1):
            genfile.write("     %d     %d     %d"     %     (1     if
genotypes[j][i]==typemap[1] else 0,1 if genotypes[j][i]==typemap[2]
else 0,1 if genotypes[j][i]==typemap[3] else 0))
        genfile.write("\n")
    samplefile.write("ID_1 ID_2 missing")
    for i in range(0,type_number):
        samplefile.write(" pheno_%d" % (i))
    samplefile.write("\n0 0 0%s\n" % (" B"*type_number))
    for i in range(len(phenotypes)):
        samplefile.write("%d %d 0" % (i,i))
        phenotype = phenotypes[i]
        for pheno in phenotype:
            samplefile.write(" %d" % (pheno))
        samplefile.write("\n")
    samplefile.close()
    genfile.close()
```

Plink 样本数据预处理

```
#!/usr/bin/python
import sys

genofile = open(sys.argv[1],'r')
phenofile = open(sys.argv[2],'r')

pedfile = open(sys.argv[3],'w')
genotypefile = open(sys.argv[4],'w')
mapfile = open(sys.argv[5],'w')

line_number=0
```

```
phenotypes=[]
for line in phenofile:
    splits = line[:-1].split(" ")
    phenotype=[]
    for sp in splits:
        phenotype.append(int(sp))
    phenotypes.append(phenotype)
print "Sample Size: %d." % (len(phenotypes))
for line in genofile:
    if line_number==0:
        splits=line[:-1].split(" ")
        for split in splits:
            mapfile.write("1 %s 0 100000\n" % (split))
    else:
        genotypefile.write("%d 1" % (line_number))
        for sp in phenotypes[line_number-1]:
            genotypefile.write(" "+str(sp+1))
        genotypefile.write("\n")
        pedfile.write("%d 1 0 0 1" % (line_number))
        pedfile.write(' '+line)
    line_number+=1
print "Done %d lines data." % (line_number)
```